

WHAT IS CLAIMED IS:

1. An improved method for a client to make a remote method call on a service, the method comprising:

obtaining an interface definition of a particular service available to remote clients,

5 said interface definition including runtime type information;

casting a generic interface class to said particular service for dynamically generating a proxy at runtime for making a remote method call on said particular service; and

when a call is made by a client on said particular service, said proxy automatically making a remote method call on said particular service by performing the substeps of:

10 converting said call made by said client into a converted call in a wire format specified in said interface definition; and

sending said converted call to said particular service using a method of transport specified in said interface definition.

15 2. The method of claim 1, wherein said generic interface class is subclassed to provide support for particular wire formats.

3. The method of claim 1, wherein said generic interface class is subclassed to provide support for particular methods of transport.

4. The method of claim 1, wherein said generic interface class may be used with a plurality of wire formats and method of transports.

20 5. The method of claim 1, wherein said step of casting a generic interface class includes casting a subclass of said generic interface class that supports a particular wire format and method of transport.

6. The method of claim 5, wherein said wire format is Simple Object Access Protocol (SOAP) and said method of transport is HyperText Transfer Protocol (HTTP).

25 7. The method of claim 1, wherein said substep of converting said call includes serializing said call into a particular wire format.

8. The method of claim 7, wherein said particular wire format includes Simple Object Access Protocol (SOAP).

9. The method of claim 1, wherein said substep of sending said converted call includes sending said converted call via the Internet.

5 10. The method of claim 1, wherein said substep of sending said converted call includes sending said converted call using the HyperText Transfer Protocol (HTTP).

11. The method of claim 1, wherein said substep of sending said converted call includes sending said converted call in the manner specified in said interface definition.

10 12. The method of claim 1, wherein said substep of sending said converted call includes receiving return values in response to said converted call.

13. The method of claim 12, wherein receiving return values includes deserializing said return values.

14. The method of claim 1, further comprising:
said proxy returning results of said remote method call to said client by performing
15 the substeps of:
receiving packets returned in response to said remote method call;
converting said packets into results in native format; and
returning said results in native format to said client.

20 15. An improved method for making a service available to remote clients, the method comprising:

creating an interface definition for a particular service, said interface definition including runtime type information;

implementing said interface definition as part of said particular service;

25 in response to a request received from a remote client on said particular service, automatically converting said request into a native call on said particular service; and

converting return values resulting from said native call on said particular service into a format appropriate for return to said remote client and sending said return values to said remote client.

5 16. The method of claim 15, wherein said step of implementing said interface definition includes installing implemented classes on a Web server.

17. The method of claim 15, wherein said step of implementing said interface definition includes registering said interface to make said particular service available to remote clients.

10 18. The method of claim 15, wherein said step of implementing said interface definition includes implementation of a dispatcher that listens for requests made on said particular service.

19. The method of claim 18, wherein said dispatcher listens for requests made in accordance with said interface definition.

15 20. The method of claim 15, wherein said step of converting said request into a native call includes deserializing packets received by from said remote client.

21. The method of claim 20, wherein deserializing packets includes deserializing Simple Object Access Protocol (SOAP) packets.

22. The method of claim 15, wherein said step of converting said request into a native call includes determining a method of said particular service to be invoked.

20 23. The method of claim 15, wherein said step of converting return values includes serializing said return values into a particular wire format.

24. The method of claim 23, wherein said particular wire format is the same format in which the request is received from said remote client.

25. The method of claim 23, wherein said particular wire format includes Simple Object Access Protocol (SOAP).

26. The method of claim 15, wherein said step of sending return values includes sending said return values via the Internet.

5 27. The method of claim 15, wherein said step of sending return values includes using a particular method of transport.

28. The method of claim 27, wherein said method of transport is the same method of transport used by said remote client to make the request on said particular service.

10 29. The method of claim 27, wherein said method of transport includes HyperText Transfer Protocol.

30. A system for making a service available to remote clients, said system comprising:

15 an interface definition for a particular service to be made available to remote clients;
a dispatcher module for listening for remote method invocations on said particular service, receiving said remote method invocations in a predetermined wire and transport format, deserializing said remote method invocations into native format, reserializing returned result values from native format into said predetermined wire and transport format, and returning said reserialized return values; and

20 an invoker module for making a native call on said particular service and returning result values from said native call.

31. The system of claim 30, wherein said invoker module returns results of said native call to said dispatcher module.

25 32. The system of claim 30, wherein said predetermined wire and transport format includes Simple Object Access Protocol (SOAP) wire format and HyperText Transfer Protocol (HTTP) transport format.

33. The system of claim 30, further comprising:
a data structure for containing deserialized information from said remote method invocation and return values from said particular service.

34. The system of claim 33, wherein said dispatcher module deserializes remote
5 method invocations into said data structure.

35. The system of claim 33, wherein said dispatcher module provides said data structure to said invoker module.

36. The system of claim 33, wherein said invoker module places return values from said native call into said data structure.

37. The system of claim 33, wherein said dispatcher module reserializes return values
10 in said data structure and returns said reserialized return values in response to said remote method invocation.

38. The system of claim 30, wherein said dispatcher module is implemented as part of a Web server module.

39. The system of claim 30, wherein said dispatcher module listens for a remote
15 method invocations in particular wire formats.

40. The system of claim 30, wherein said dispatcher module listens for remote method invocations received via the Internet.

41. The system of claim 30, wherein said dispatcher module listens for HyperText
20 Transfer Protocol (HTTP) posts containing Simple Object Access Protocol (SOAP) requests.

42. A method for making a service available to remote clients, said method comprising:
developing an interface definition to a service to be made available to remote clients;

implementing said interface definition to create an implemented interface to said service, said implemented interface listening for requests on said service;

in response to receipt of a remote method call said service, deserializing said remote method call into native format;

5 invoking said service by making a native call on said service in native format; and
reserializing results of said native call on said service and returning said reserialized results in response to said remote method call.

10 43. The method of claim 42, wherein said step of implementing said interface definition includes registering said implemented interface to make said service available remotely.

44. The method of claim 42, wherein said interface definition includes an invokable interface; said invokable interface enabling remote clients to cast an instance of said interface for purposes of making a remote method call on said service.

15 45. The method of claim 42, wherein said implemented interface includes a dispatcher module for handling remote method calls on said service.

46. The method of claim 42, wherein said step of deserializing said remote method call includes using the same wire format used to serialize the request.

47. The method of claim 42, wherein said step of deserializing said remote method call includes deserializing a Simple Object Access Protocol request.

20 48. The method of claim 42, wherein said step of deserializing said remote method call includes deserializing said remote method call into a data structure.

49. The method of claim 48, wherein said step of invoking said service includes passing said data structure to said service.

50. The method of claim 42, wherein said step of returning said serialized results includes returning said serialized results using the same method of transport used to send said remote method call

5 51. A system enabling a client to invoke a remote service, the system comprising:
a generic interface class providing client-side support for invocation of a remote
service having an available interface definition;
an interface object dynamically generated from said generic interface class at runtime
which communicates with said remote service; and
a serialization interface which serializes data into a particular wire protocol for
10 invocation of said remote service and deserializes return values received from said remote
service.

52. The system of claim 51, wherein said generic interface class is subclassed to
provide support for particular wire formats.

15 53. The system of claim 51, wherein said generic interface class is subclassed to
provide support for particular methods of transport.

54. The system of claim 51, wherein said generic interface class may be used with a
plurality of wire formats and method of transports.

20 55. The system of claim 51, further comprising:
a subclass of said generic interface class providing support for a particular wire
format and method of transport.

56. The system of claim 55, wherein said wire protocol is Simple Object Access
Protocol (SOAP)

57. The system of claim 51, wherein communication with said remote service
includes sending HyperText Transfer Protocol (HTTP) posts.

58. The system of claim 51, wherein communication with said remote service includes communication via the Internet.

59. The system of claim 51, wherein communication with said remote service includes receiving responses from said remote service.

5